DATA PROCESSING AND DATA BASE MANAGEMENT INNOVATIONS--THE PANEL STUDY OF INCOME DYNAMICS

Paula A. Pelletier and Michael A. Nolte

INTRODUCTION

The Panel Study of Income Dynamics is now in its eleventh year. Since 1968 it has followed, conducted annual interviews with, and processed information from a national sample of about 5,000 families. The content of this study is largely economic, with considerable detail on family income. The PSID database currently contains ten years of information for 6007 families and 21000 individuals; a new wave of family information (500 variables, a logical record length of around 1000 bytes) and individual information (20 variables, logical record length about 40 bytes) is added each year. Because the PSID follows families over time, data collection, processing and management procedures must cope with the complexities of family composition changes while remaining within an annual processing schedule.

Data consistency within each wave and over time is important since PSID data are used in estimation of micro-level behavioral relationships. A great deal of person and computer time is spent doing cross-file consistency checks. When errors are detected, staff members return to the original interview schedules to determine what corrections should be made. These corrections are then incorporated into the database.

PSID data processing procedures have evolved slowly over the past ten years of data collection. One reason for this has been the caution exercised by the study staff in replacing proven procedures with new and untried ones. The analyst's need for cross-year consistency within the data has also proven a powerful argument against over-hasty adoption of new procedures. Furthermore, the positive feedback received from external users of PSID data who are pleased with its internal consistency and the accessibility of all of its data in a rectangular format has argued against major structural changes. During the past two years however, change on a large scale in internal processing procedures has been forced upon the PSID staff. The two most important factors leading to change are the growing size and complexity of the PSID database as well as the personnel costs associated with its processing.

The Complexity of Family Structure

Over the years the PSID data management staff have become as concerned with the accurate representation of family composition as they have been with the accurate assessment of family economic structure. Since many of the derived income measures, (e.g., the ratio of family income to family needs) are based on the number of people living in the household, it is essential that the correct individuals appear on the interview schedule. This would seem to be a simple task, but since the PSID follows all new families that split off from the original 1968 root (sample) family, it is not uncommon to find individuals who move between different sample families. Other changes in family structure, such as having sample members from two different 1968 root families marry and start their own new family unit (three such cases exist at present), do much to complicate all aspects of PSID data handling procedures.

INNOVATIONS

Developmental work in all phases of the survey research process is being carried out by various units of the ISR. Efforts currently underway include the improvement of telephone interviewing methodology (on-line interviewing as well as random computer selection of telephone numbers within area code); creation of on-line systems to maintain up-to-date respondent recontact information and to monitor field operations; extension of OSIRIS to include a structured file processing capability; improvement of OSIRIS data analysis capabilities; development of report generation and graphics display software; and finally, creation of a sophisticated general purpose direct data entry system. The PSID is currently supporting development in three of these areas: automation of respondent recontact procedures, use of direct data entry, and implementation of a hierarchically structured database.

RESPONDENT RECONTACT

Cover Sheet Control

As each wave of PSID data collection is begun, blank interviews, together with fresh coversheets for the current year (and, if necessary, coversheets from previous years) are transmitted to interviewers in the field. When a completed interview arrives in Ann Arbor, its coversheet is removed for storage in a separate physical location. Before being stored however, certain data from the cover sheet are entered in a direct access disk file. This control file allows the study staff to obtain immediate information on the status of the interviewing process; it also allows the field office to monitor the progress of interviewers and to take corrective action if deficiencies in their techniques are noted. Pointer fields in the control file allow it to serve two additional purposes. The file is linked to ISR business office files to obtain interviewer expense data for comparison with interviewer performance. The control file is also linked to the address file maintained by the PSID study staff. This link allows the PSID staff to determine, for payment purposes, which respondents have actually been interviewed in a given wave of data collection.

The Address File

A panel study depends on successful respondent recontact--no reinterviews, no panel. Thus, keeping track of respondents over time is a task of paramount importance. In the PSID, the necessity for keeping track of both families and individuals makes this task much more complex. The solution currently employed by the PSID is to maintain a continuously updated direct access file of family addresses.[1]

The address file contains the name of the head of household as well as other identifying information for each family currently in the PSID sample. This file is updated at the beginning of each year, when respondent families send in post-cards with their current address. It is also updated during the interviewing cycle as interviewers report new addresses, new families created from old families, and the disappearance of old families. Use of the address file falls into three main categories: 1.) generation of labels containing identification number or address information for coversheets and respondent mailings; 2.) linkage to the University of Michigan accounting system for payments to respondents; 3.) linkage to the cover sheet control process.

Use of the address and field control files has greatly simplified the administrative effort necessary to maintain contact with the PSID sample. The possibility of human error can now be confined to the point of data entry. Files remain current, since they can be updated on a daily basis. Reports can be generated on the basis of immediate requirements and need not await some predetermined point in the processing cycle.

DIRECT DATA ENTRY

In essence, the direct data entry (DDE) system currently under development at the ISR provides a direct link between coder and machine. Data can be entered via computer terminal into a database from which a software interface in turn produces an OSIRIS data file. The DDE system is a general purpose system; it can be modified to suit most (if not all) data collection methods, and can be used for more than onestudy at a time.

Advantages of DDE

The versatility of DDE can perhaps best be seen by comparing it with the more traditional "pencil and paper" method of survey data coding. Codesheets and punchcards are eliminated, since a coder's evaluation of an interview is put directly into machine readable format. Because the data are machine readable, checks for wild codes and inconsistencies can be built into the entry process, and incorrect coder decisions can be immediately flagged for correction. Interview skip patterns are built into the programmed instructions for each project; this allows the computer to check forward consistency, and to ensure that coding at any point in an interview is consistent with the coding at any previous point. Certain types of data aggregation, especially those that are highly dependent upon accurately entered data, can also be built into the entry process. The net result of the DDE process is clean data, available for further processing or analysis within a relatively short time frame.[2]

STRUCTURED FILE DEVELOPMENT

In the past three years the PSID has outgrown the processing system which worked well during the first seven years. The two primary factors leading to the system's obsolescence have been the increasing size and complexity of the PSID files and the person hours associated with processing and maintenance. For example, our Ninth Year Family-Individual file resides on five 1600 BPI tape reels, with an N of 21000 and a logical record length of 9381. Processing the data file is increasingly constrained by hardware and software limitations. The decision to use the OSIRIS structured file capability arose out of the need to simplify the processing task.

The PSID Database

Before moving to a structured file system, the PSID data management staff spent a considerable amount of time developing an abstract structure, or schema, that would provide for optimal organization of the study's data. Since any proposed structure had to satisfy all of the data retrieval needs of the PSID researchers, the schema required would minimally have to provide for the retrieval of: a current year family file, a cross-year family file, a current year family-individual file, and a cross-year family-individual file. A data base organized around the construct of family history promises to provide all of the above.

The PSID database is bounded by three elements: family, individual, and time. Within each year's data, there exists a natural hierarchy since each set of individual records is linked to one family record. However, the introduction of the time element destroys the usefulness of the simple family-individual hierarchy. As has been mentioned previously, a family may experience a great deal of structural change over time: sample members leave and start their own homes (and may later reenter their original family), new sample members are born in, and non-sample members move in and out. Because these family changes are so widespread and because individuals frequently move between sample families over time, it is often difficult to link a set of individuals who are living together as a family in the current year to one common family history. That is, a family-individual hierarchy comprised of all the past years of family data followed by past individual records of all individuals currently residing in that family is a schema that will not work. Individuals who are living together now may have been members of different families in previous years. Linking them all to one set of supposedly shared past family records would be inaccurate. Also, when there are radical changes in family structure over time it is often quite difficult to devise a rule that determines which branch family is most representative of the 1968 root family line.

Developing the Family History Concept

The notion that the PSID follows families over time has been the source of much confusion. From a data management perspective, the PSID does not follow families over time; rather, it follows individuals who aggregate themselves into families at points in time. The study collects family data each year and introduces new sample members who are born into a family into the database, but to the data processing staff, "family" is a construct imposed on a collection of individuals who happen to be living together at a given point in time. The study staff think that this formulation has implications for many longitudinal studies using multiple level data.

Two examples of cases that would not be adequately represented by a simple family-individual hierarchy follow: The first is a situation where sample daughter X moves between her parental family and the family of an older sister. Both of these families are sample families with separate records on our file. In order to perform an accurate analysis of daughter X's data, she must be linked to all of the families of which she was a member. A second example is that of a husband and wife who split up into separate families for a few years and later recombine. In order to have an accurate assessment of a person's economic well-being over time, the individual must be linked to all of his or her previous family records, even if that family line has been abandoned.

The construct of family history was developed to deal with this problem of linking individuals to their correct past family records. A family history is simply all the yearly family records that one individual has moved through over time. In most PSID sample families, the majority of respondents share similar family histories. However, in structurally unstable family units many different family histories may be present. Figure 1 presents an example of a family that has undergone a moderate amount of change in four years. This family (1968 ID = 1010) will be used as our example on the following pages.

In Figure 1, note that as of 1975, family '1017' has 2 family histories for 4 individuals. Individuals 01,02 and 31 have the same family history (1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017). The family history of individual 07 reflects membership in split-off family 4040 during 1969 and 1970 (1010, 4040, 4041, 1013, 1014, 1015, 1016, 1017).

The OSIRIS Structured File Facility

In OSIRIS IV a structured dataset is built from one or more individual rectangular files via SBUILD. Each rectangular dataset becomes one or more groups in the structured dataset. Later, when a structured dataset is retrieved as an input to an OSIRIS program, the user provides rules to govern how the groups are to be arranged to create rectangular records called entries. These rules are supplied within an OSIRIS procedure called ENTRY, using a set of instructions called the Entry Definition Language . Thus structured files in OSIRIS IV are created by SBUILD and accessed by analysis programs according to the rules of the Entry Definition Language.

When creating a structured file within OSIRIS IV the user describes the hierarchical relationships within the data to the SBUILD program by means of structure definition statements. This logical description of the hierarchy is called a schema. An actual occurrence of a schema, analogous to a case in a rectangular file, is known as an instance.

Instead of pointers which link records within a disk data set, OSIRIS "links" its records to each other through sort fields generated by the SBUILD program. These sort fields contain identification variables. Each record contains, as well as its own unique ID, the ID of every other record in the path from it to the root of the tree structure that describes its instance. The structured file produced by SBUILD is arranged in ascending order on these sort fields.

A disadvantage of this sequential access method is that it is necessary to read all

Figure 1: Chart of 1968 Seed (root) Family '1Ø1Ø'

Year	Main Fam.	Per-son#	Sex	Age	Split -off Fam.	Per-son#	Sex	Age
1968	1Ø1Ø	Ø1	m	45				
		Ø2	f	43				
		Ø3	m	25				
		Ø4	m	24				
		Ø7	f	11				
1969	1Ø11	Ø1	m	46	4Ø4Ø	Ø4	m	25
		Ø2	f	44		Ø7	f	12
		Ø3	m	26				
197Ø	1Ø12	Ø1	m	47	4Ø41	Ø4	m	26
		Ø2	f	45		Ø7	f	13
		31	f	Ø1				
1971	1Ø13	Ø1	m	48	4Ø42	Ø4	m	27
		Ø2	f	46				
		31	f	Ø2				
		Ø7	f	14				
1972	1Ø14	Ø1	m	49	4Ø43	Ø4	m	28
		Ø2	f	47				
		31	f	Ø3				
		Ø7	f	15				
1973	1Ø15	Ø1	m	5Ø	4Ø44	Ø4	m	29
		Ø2	f	48				
		31	f	Ø4				
		Ø7	f	16				
1974	1Ø16	Ø1	m	51	4Ø45	Ø4	m	3Ø
		Ø2	f	49				
		31	f	Ø5				
		Ø7	f	17				
1975	1Ø17	Ø1	m	52	4Ø46	Ø4	m	31
		Ø2	f	5Ø				
		31	f	Ø6				
		Ø7	f	18				

preceding records in order to retrieve a particular record, and to copy the entire file to add or delete a record. It should be stressed, however, than this disadvantage is more than outweighed by the consideration of storage costs. At the present time, direct access storage is not cost effective for large databases, since cost/byte for tape storage is several orders of magnitude less than that for disk storage. Furthermore, the direct access storage devices of many computing installations lack the physical capacity to encompass an extremely large database such as the PSID. An additional consideration is that social science statistical analysis usually involves accessing an entire data file or a significant subset of one; in many cases sequential processing of a tape file may be more efficient than sequential processing of a direct access file.[3]

As a data record is input to SBUILD during the build process, sort fields are attached to the beginning of that record. After all records are processed, the entire file is sorted. This sorting produces a file of instances that is arranged in an order that approximates a left to right preorder traversal of the tree structure. Starting with the root node, the nodes are traversed according to a set of prioritized "directions": 1.) down the left-most branch, 2.) across (left to right) any sibling nodes at the same level, and 3.) up one level to the next node.[4]

Figure 2 presents a PSID instance as it

Fig. 2: OSIRIS Sort Fields (PSID Database).

GRP # Level	75ID 1	FH# 2	CONST 3	68ID 3	PER # 3	CONST 4	Data
Ø1	1Ø17						75 F
Ø2	1Ø17	1	Ø1				68 F
Ø3	1Ø17	1	Ø2				69 F
Ø4	1Ø17	1	Ø3				7Ø F
Ø5	1Ø17	1	Ø4				71 F
Ø6	1Ø17	1	Ø5				72 F
Ø7	1Ø17	1	Ø6				73 F
Ø8	1Ø17	1	Ø7				74 F
68	1Ø17	1	Ø7	1Ø1Ø	Ø1	Ø1	68 I
69	1Ø17	1	Ø7	1Ø1Ø	Ø1	Ø2	69 I
7Ø	1Ø17	1	Ø7	1Ø1Ø	Ø1	Ø3	7Ø I
71	1Ø17	1	Ø7	1Ø1Ø	Ø1	Ø4	71 I
72	1Ø17	1	Ø7	1Ø1Ø	Ø1	Ø5	72 I
73	1Ø17	1	Ø7	1Ø1Ø	Ø1	Ø6	73 I
74	1Ø17	1	Ø7	1Ø1Ø	Ø1	Ø7	74 I
75	1Ø17	1	Ø7	1Ø1Ø	Ø1	Ø8	75 I
68	1Ø17	1	Ø7	1Ø1Ø	Ø2	Ø1	68 I
69	1Ø17	1	Ø7	1Ø1Ø	Ø2	Ø2	69 I
7Ø	1Ø17	1	Ø7	1Ø1Ø	Ø2	Ø3	7Ø I
71	1Ø17	1	Ø7	1Ø1Ø	Ø2	Ø4	71 I
72	1Ø17	1	Ø7	1Ø1Ø	Ø2	Ø5	72 I
73	1Ø17	1	Ø7	1Ø1Ø	Ø2	Ø6	73 I
74	1Ø17	1	Ø7	1Ø1Ø	Ø2	Ø7	74 I
75	1Ø17	1	Ø7	1Ø1Ø	Ø2	Ø8	75 I
68	1Ø17	1	Ø7	1Ø1Ø	31	Ø1	68 I
69	1Ø17	1	Ø7	1Ø1Ø	31	Ø2	69 I
7Ø	1Ø17	1	Ø7	1Ø1Ø	31	Ø3	7Ø I
71	1Ø17	1	Ø7	1Ø1Ø	31	Ø4	71 I
72	1Ø17	1	Ø7	1Ø1Ø	31	Ø5	72 I
73	1Ø17	1	Ø7	1Ø1Ø	31	Ø6	73 I
74	1Ø17	1	Ø7	1Ø1Ø	31	Ø7	74 I
75	1Ø17	1	Ø7	1Ø1Ø	31	Ø8	75 I
Ø2	1Ø17	2	Ø7				68 F
Ø3	1Ø17	2	Ø7				69 F
Ø4	1Ø17	2	Ø7				7Ø F
Ø5	1Ø17	2	Ø7				71 F
Ø6	1Ø17	2	Ø7				72 F
Ø7	1Ø17	2	Ø7				73 F
Ø8	1Ø17	2	Ø7				74 F
68	1Ø17	2	Ø7	1Ø1Ø	Ø7	Ø1	68 I
69	1Ø17	2	Ø7	1Ø1Ø	Ø7	Ø2	69 I
7Ø	1Ø17	2	Ø7	1Ø1Ø	Ø7	Ø3	7Ø I
71	1Ø17	2	Ø7	1Ø1Ø	Ø7	Ø4	71 I
72	1Ø17	2	Ø7	1Ø1Ø	Ø7	Ø5	72 I
73	1Ø17	2	Ø7	1Ø1Ø	Ø7	Ø6	73 I
74	1Ø17	2	Ø7	1Ø1Ø	Ø7	Ø7	74 I
75	1Ø17	2	Ø7	1Ø1Ø	Ø7	Ø8	75 I

would appear within an OSIRIS structured dataset. For simplicity each record is identified only by its node number.

Retrieval.

Once a data set has been transformed into a hierarchical structure the user next has to ascertain how to retrieve the data. This is done via special ENTRY retrieval instructions which the user includes with the OSIRIS accessing program. These instructions are provided by default through the original SBUILD process; they may be overridden to suit the needs of the analyst.

In order to understand how OSIRIS IV accomplishes retrieval of structured file records, the user should consider the output records resulting from SBUILD. These records are stored sequentially, whether on magnetic tape or direct access storage device. The records themselves are of varying length (VBS format) and composition; for example, a family level record of length 800 may be followed by an individual data record of length 40. As each record is read, ENTRY examines that record's identification fields (i.e., its sort fields) to determine the group and level with which the record is associated. ENTRY then makes use of an algorithm in which the sort fields of the current record are compared to the sort field (now saved) of the previous record. If a change in ID field has occurred, the program recognizes that it has reached the end of a "branch" and has come upon a "leaf" (a leaf is a node in a tree structure having no subsidiary nodes).

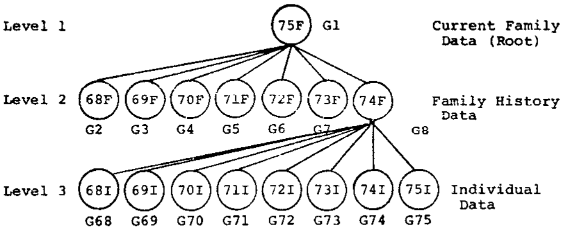
When the end of a leaf is signalled -- that is, when the identification fields change at the unit of analysis level -- all the information associated with the branch on which that leaf appears is "evaluated". The ENTRY procedure now has to decide what to do with the data associated with the leaf. Depending on the criteria specified by the user within the Entry Definition Language, the data will or will not be passed to the OSIRIS accessing program. After the pass/no-pass evaluation is made and conditions of the entry have been met, new data (indicated by changed sort fields in the new record) replace the previously saved data at the appropriate level.

Instructions within ENTRY allow the user to specify when an entry is to be evaluated, what information should be present before ENTRY passes a complete record to the accessing program, and which identification fields determine a particular leaf. By manipulating the Entry Definition Language, the user can impose a new structure on a particular structured file without rebuilding it. ENTRY is thus an extremely powerful tool for the analyst.

OSIRIS and the PSID Database.

Figure 3 presents the logical structure, i.e., the schema of the PSID analysis data base (this example is based on the 8-year Family-Individual file). At the top, or root, is the most current year's family record; level two contains the family history records; and level 3 contains the individual/year data. Each of these nodes represents a different record type, i.e., a distinctly different group of variables. All the individuals living in family X during year Y have their past individual records (level 3) linked to the family records of which they have been part in the past (level 2) and to the current family record (level 1).

Figure 3: Schema -- logical structure of PSID data.



Note group numbers -- since it is possible to have multiple groups at the same level, group numbers are necessary to identify each group uniquely.

Within the SBUILD setup the user must define ID and/or link variables for each group. These ID numbers are the basis of the sort fields which are attached to the front of each record.

Figure 4: Instance -- an actual occurrence of the PSID schema (1975 family '1017')

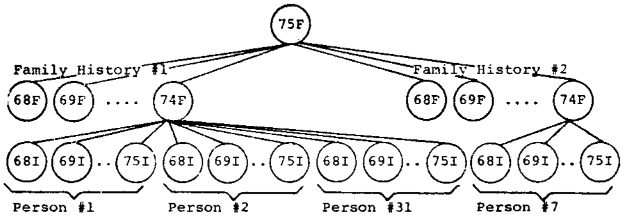


Figure 2 is an actual representation of what the sort fields would look for a sample family '1017' in 1975. Each record, as well as having the identification field appropriate to its own level, also has the identification fields of any records above it in the specific instance hierarchy.

Note that two digit constants are generated by the program to distinguish between groups at the same level sharing identical sort ID's. These constants are not considered part of the ID at the level to which they apply, but rather they are treated as part of the ID of the next lower level. For example, groups 02 through 08 as level 2 records have the same 1975 ID and family history number sort fields. The constants generated by the program to distinguish among them are not level 2 sort fields but rather they apply to a level 3 identification. Similarly, the constants generated at the individual/year level are not part of the level 3 sort fields but rather that of level 4.

Retrieval of PSID Data

To retrieve any subset of the database the analyst must first conceptualize the appropriate sub-schema for analysis (a sub-schema is that section of the logical data structure required for a given piece of analysis). The user gives the correct retrieval instructions via ENTRY in order to get the correct sub-instances. Figure 5 outlines the basic ENTRY instructions necessary to implement typical PSID sub-schemas. The Entry Definition Language defines the sub-schema; the ENTRY procedure passes each sub-instance to the analysis program upon execution.[5]

Figure 5: Accessing the PSID
1968-1975 Database

Analysis Subset (sub-schema)	ENTRY Instructions
Cross-year Family-Individual	UNIT=3 G1+G2+..+G8+G68+..+G75
Cross-year'Family	UNIT=1 G1+G2+...+G8
Current-year Family	UNIT=1 G1
Current-year Family-Individual	UNIT=3 G1+G75
Person-year	UNIT=4 G1+G2+..+G8+ (G68/G69/../G75)

CONCLUSIONS

In this paper we document the PSID experience in developing a sequentially ordered hierarchical database. With each additional year of data the PSID database has become larger and more complex. Original data management techniques were made obsolete by hardware and software limitations. In the move to a new database structure we have learned lessons which may prove interesting to others engaged in planning or designing database systems for longitudinal studies. The PSID staff have discovered from experience that:

1.) Designing a database that will fulfill both the requirements of the analysis staff and the data management personnel may require a large amount of time and energy.

2.) In the initial stages of a longitudinal study, it is important to retain maximum flexibility in storing collected data. When data are collected on a variety of levels, all interrelationships (and hence the real structure of the data) may alter over time in ways which were unexpected at the start of the study. Any database must be flexible enough to accommodate these new levels of complexity.

3.) Close contact with the research and analysis staff (as well as outside users) is necessary at all points in the development process. It is important to take analysis needs, both present and anticipated, into account when developing a database structure.

4.) The database access and storage techniques must be examined thoroughly before implementation. Maximum retrieval efficiency and the elimination of redundancy in record storage should be design aims.

5.) Procedures for administrative support (e.g., control of interviews and coversheets, maintenance of contact with respondents, editing and coding of data) should be designed, tested and implemented before the data collection process begins. These procedures should operate with maximum accuracy and minimum human effort. A high degree of accuracy will cut processing time and improve control of the sample; minimization of human effort will cut personnel costs.

Study planners should be aware that analysis interests may change during the course of their study. Also, planners should recognize that computing technology will continue to become cheaper and more efficient over the long term. Design of data collection and analysis procedures should allow for modifications to allow study staff to take advantage of improvements in the computing environment. All in all, given the plethora of design parameters, it may be necessary to forego the concept of a "final" database design in favor of a design in which the need for evolution and revision is assumed from the start.

[1] It should be noted that each year's processing cycle produces a tape file containing information (including name) on all individuals within each family. This file is used subsequently in production of the next year's batch of coversheets. It is not linked to the address file.

[2] Transcription of data from the PSID interview schedule is divided into two separate processes, editing and coding. During editing, year to year changes in family membership are accounted for, discrepancies within the interview are rectified and numeric variables (e.g., total taxable income) are calculated and written to worksheets. Coding, which follows editing, is the assignment of numeric codes to respondent answers.

[3] See Robert F. Teitel, "A Relational Database Approach to Social Science Computing" (1977).

[4] The method used by OSIRIS IV to read a structured data file while retaining its hierarchical pattern is the preorder traversal mentioned above.

[5] Certain additional instructions may be necessary to allow for such circumstances as missing data due to non-response in one or more years. These instructions have been eliminated in our examples for the sake of simplicity.

REFERENCES

Date, C.J., An Introduction to Data Base Systems, Addison Wesley (1976)

Kronke, David, Fundamentals of Database Systems, Science Research Associates (1977)

Martin, James, Principles of Data-Base Management, Prentice-Hall (1976)

Panel Study of Income Dynamics, Procedures and Tape Codes, (Waves 1-10), Institute for Social Research (1978)

Survey Research Center Computer Support Group, OSIRIS IV User's Manual, Institute for Social Research (1977)